Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms*

O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat, Z. Zhang

INRIA Sophia-Antipolis

2004 Route des Lucioles

06565 Valbonne Cedex

Note 1: The citation of this paper is:

O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat, and Z. Zhang, "Quantitative and Qualitative Comparison of Some Area and Feature-based Stereo Algorithms", In Wolfgang Forstner and Stephan Ruwiedel, editors, *Robust Computer Vision: Quality of Vision Algorithms*, pages 1–26. Wichmann,Karlsruhe, Germany, 1992.

Note 2: This version is reconstituted from Zhang's old archive, and many images were missing.

Abstract

Evaluating the performance of stereo algorithms, both in terms of robustness and precision, is of critical importance as they become more amenable to practical applications. It is, however, a hard task because no unified testbed exists. In this paper, we use the algorithms that have been developed at INRIA in recent years to propose a number of methods that can be used to achieve this task and evaluate the appropriateness of algorithms for given applications.

*Support for this research was partially provided by ESPRIT P2502 (VOILA), the CNES VAP contract, the EUREKA Prometheus Contract and a Defense Advanced Research Projects Agency contract (at SRI int.).

1 Introduction

Passive stereo is becoming more and more readily applicable to domains where 3D sensing is required, such as robot navigation or cartography. This evolution is taking place because

1. algorithms have matured,

2. hardware has improved to the point where their implementation in real time is now possible at low cost.

Of course, there exist many different stereo algorithms and the question of choosing one for a specific application is not an easy one. These algorithms can roughly be classified in two categories [**?**]:

1. **Feature Based.** These algorithms extract features of interest from the images, such as edge segments or contours, and match them in two or more views. They are fast because only a small subset of the image pixels are used, but may fail if the chosen primitives cannot be reliably found in the images; furthermore they usually only yield very sparse depth maps which can nonetheless be interpolated [**?**].

2. **Area Based.** In these approaches, the system attempts to correlate the grey levels of image patches in the views being considered, assuming that they present some similarity. The resulting depth map can then be interpolated. The underlying assumption appears to be a valid one for relatively textured areas; however, it may prove wrong at occlusion boundaries and useless within featureless regions.

   Alternatively the map can be computed by directly fitting a smooth surface that accounts for the disparities between the two images. This is a more principled approach since the problem can be phrased as an optimization one; however, the smoothness assumptions that are required may not always be satisfied.

These techniques have their strengths and weaknesses and it is difficult to assess their compared merits since few researchers work on similar data sets and very few quantitative evaluations of the results have been performed. However, one can get a feel for the relative performance of these systems from the study by Güelch [**?**]. In this work, the author has

assembled a standardized data set and sent it to 15 research institutes across the world. Having developed several new algorithms at INRIA since then, we have tested them in the same spirit and present the results in this paper. In the following section, we describe these algorithms. We then show their behaviour on real data and, finally, use synthetic data to better quantify their performances.

2 The Algorithms

We have developed both area-based stereo algorithms and feature-based ones. The first two that we describe, one relying on correlation and the other on chained contours, are binocular. The next two are trinocular and use segments and curves as their primitives.

2.1 Correlation

A number of correlation-based algorithms attempt to find points of interest on which to perform the correlation. This approach is justified when only limited computing resources are available, but with modern hardware architectures it becomes practical to perform the correlation over all image points and retain only matches that appear to be "valid." The hard problem is then to provide an effective definition of what we call validity and we will propose one below.

Correlation scores are computed by comparing a fixed window in the first image to a shifting window in the second. The second window is moved in the second image by integer increments and an array of correlation scores is generated for integer disparity values. To compute the disparity with subpixel accuracy, we fit a second degree curve to the correlation scores in the neighborhood of the maximum and compute the optimal disparity by interpolation.

As shown by Nishihara [**?**], the probability of a mismatch goes down as the size of the window and the amount of texture increase. However, using large windows leads to a loss of accuracy and the possible loss of important image features. We choose to consider as acceptable only those results for which we get the same result by reversing the roles of the two images, thereby greatly reducing the probability of error even when using very small windows (down to 3x3 for textured outdoor scenes). In fact, the density of such

consistent matches in a given area of the image appears to be an excellent indicator of the quality of the stereo matching. An occasional "false positive" (a pixel for which the same erroneous disparity is measured when matching both from left to right and right to left) may occur. But, except in the presence of repetitive patterns, we have never encountered a situation that gave rise to a large clump of such errors.

5.0cmima/Rocks1.ps 5.0cmima/Rocks2.ps 5.0cmima/Rocks3.ps

Three images of a rock scene.

5.5cmaima/RocksDisp2.ps 5.5cmbima/RocksDisp3.ps

14cm (a) The disparity map computed using windows of the first two images of Figure 2.1. (b) The merger of the maps derived from matching images 1 and 2 and images 1 and 3 after removal of the isolated points. The black areas are those for which no valid matches were found. Elsewhere, the lighter regions are those that are further away.

Other stereo systems (e.g. [**?**] and the contour based algorithm of section 2.2) include a validity criterion similar to ours but use it as only one among many others. In our case, because we correlate over the whole image and not only at interest or contour points, we do not need the other criteria and can rely on density alone. However, our validity test depends on the fact that it is improbable to make the same mistake twice when correlating in both directions and can potentially be fooled by repetitive patterns, which is a problem we have not addressed yet.

We have modeled this behaviour using synthetic data [**?**] and we have also tested the algorithm extensively on outdoor scenes such as the one of Figure 2.1. The resulting disparity maps are shown in Figure 2.1. Note that no answer, rather than a wrong answer, is given outside of the textured regions. Elsewhere the average difference between the computed disparities and hand-measured ones is .7 pixels, as will be discussed in section 3. As suggested by many researchers, [**?**, **?**, **?**] among many others, more than two images can and should be used whenever practical. By combining views and performing some simple filtering on the result, we can improve the results and return very dense maps such as the one of Figure 2.1 (e) and (f). We have also implemented adaptative

smoothing methods that yield cleaner surfaces while preserving depth discontinuities. For a more complete description of these techniques, we refer the interested reader to previous publications [**?**, **?**].

In Figure 2.1 we show a synthetic stereo pair generated from a known elevation map and an aerial view. Because the terrain is somewhat smoother, the average error is here of .5 pixels for 7x7 windows.

5.0cmaima/Alv1.ps 5.0cmbima/Alv2.ps 5.0cmcima/AlvGrid.ps

14cm(a)(b) A synthetic stereo pair generated using an aerial view and an elevation map of the area. (c) The computed elevation map using a 7x7 window shown as a mesh. The average precision of the computed disparities is .5 pixels.

This can be compared to the results of the correlation-based system described by Hannah [**?**] that "won" the Güelch study [**?**]. This system yields precisions that are better than half a pixel on average but only for a very small fraction, typically less than 1%, of the image points that are the ones the ones for which correlation works best. Our algorithm is slightly less precise but yields much denser maps; the vast majority of the points are matched in textured scenes.

Our implementation on a SPARCstation 2[1] runs in under 3 minutes on 256x256 images for any window size and a disparity range of 50. Furthermore, correlation is a very regular algorithm that can be implemented in hardware [**?**] if speed is required. It also easily parallelizable and our Connection Machine[2] implementation yields results in a few seconds.

2.2 Matching chains of contours

We now turn to a binocular contour-based stereo algorithm and concentrate on the *correspondence problem* (matching). A complete description will be found in [**?**]. To solve this problem, similarity information is of primary use. Additional constraints are needed for the purpose of disambiguation. Apart from the different image features that are used, existing matching schemes differ principally by the way in which similarity information

[1]Trademark: SUN Inc.

[2]Trademark: TMC inc.

and various constraints are taken into account. Moreover, in order to design an efficient algorithm, one should consider the noisy nature of available information and the discretization effects in digital imagery. In our case, to allow for the presence of noise, more emphasis is put on the use of global constraints than on the similarity between contour points.

2.2.1 The Constraints

In this algorithm, the following constraints are explicitly used:

1. epipolar geometry;
2. figural continuity [**?**];
3. disparity gradient (DG) limit [**?**][**?**];
4. uniqueness [**?**].

In fact, the last three constraints are related. The figural continuity derives from the fact that connected contour points usually originate from the same physical surface and thus can serve as a global support for consistency checking. The DG limit determines the ability to treat sloping surfaces. In principle, the bigger the allowed DG , the better sloping surfaces can be dealt with. However, applying the uniqueness constraint implies small DG limits, since pixels have physical size. As has been noted in [**?**], smaller DG limit provides better disambiguating power.

In practice, contour chains corresponding to steeply sloping physical surfaces are very rare. Furthermore, such surfaces produce greatly deformed image features for which, in most cases, no corresponding feature can be found. Therefore, we suggest using small DG limits. The ordering constraint for two connected points on a same epipolar line is implied by the DG limit being less than 1.

2.2.2 The algorithm

The control structure of this algorithm is a coarse-to-fine one. At each level five steps are performed: (1) candidate searching, (2) score computation, (3) match validation, (4) noise suppression and chain fractioning, and (5) disparity interpolation.

Note that both score computation and match validation are iterative processes.

Candidate searching Given a point to be matched, the search takes place along the epipolar line within a disparity range, either estimated (at the coarsest level) or predicted (at finer levels). Points encountered in the search region are then submitted to similarity examination and those satisfying a predefined criterion are retained as candidates.

Computing scores To select the best candidates, the method relies primarily on the disambiguating power of the constraints. The score of each candidate match for a given point is the sum of the support it has received from its neighbors; each neighbor's support is the maximal one its candidates can give.

The support is determined by comparing the underlying disparity gradient to the predefined DG limit (DG_0) as follows:

Underlying	Support from	
DG	Candidate match	Validated match
$= 0$	1	2
$< DG_0$	2	4
$> DG_0$	0	0

In other words, only within-DG-limit matches give support; a validated match and a match (candidate or validated) having minimal DG are favored and give more support. In the case where the neighboring point has more than one within-DG-limit candidate, the maximal support is taken into account.

Due to the discrete nature of image coordinates and thus the 1 pixel precision in contour detection, the DG is compared to $1/dist + DG_0$, rather than directly to DG_0 ($dist$ is the distance between the two involved neighboring points lying on the same contour chain).

Validating matches A candidate match is validated if both points involved are at the head of the candidate list of the other in terms of score. To enforce the uniqueness constraint, matched points are not considered anymore as candidate of any unmatched point.

Noise suppression and chain fractioning Noisy matches are those which violate the figural continuity and DG limit constraints, translated into disparity continuity in the algorithm. However, only one-sided disparity continuity is required. This is because in contour chaining, it may happen that two contour points resulting from two different surfaces are connected and linked. In such a case, the disparity continuity can be guaranteed at one side but not at the other. Contour chains are fractioned at such positions to ensure that each contour chain has a smoothly varying disparity. After this operation, each resulting contour is more likely to belong to a single surface which generally helps higher level processing and interpretation.

Disparity interpolation Once noisy matches have been eliminated and contour chains with discontinuous disparity fractioned, the interpolation of disparity can be performed safely for unmatched points in a contour.

2.2.3 Results and comments

5.0cmaima/RocksMa00.ps 5.0cmbima/RocksMa90.ps 5.0cmcima/RocksMa09.ps
5.0cmdima/RocksCor00.ps 5.0cmeima/RocksCor90.ps 5.0cmfima/RocksCor09.ps
5.0cmgima/RocksLuc00.ps 5.0cmhima/RocksLuc90.ps 5.0cmiima/RocksLuc09.ps
14cm(a) The reconstructed 3D contours for the rock scene of image 2.1 projected on the first camera in (a), seen sideways in (b) and seen from above in (c). For comparison's sake, in (d) (e) (f) we show the correlation results in the same fashion and in (g) (h) (i) the results of the trinocular algorithm of section 2.4.

In Figure 2.2.3, we show the reconstructed 3D contours for the rock scene of image 2.1. The results are quite dense and sufficient for obstacle detection. In this example, the DG limit is fixed at 0.2. The neighborhood radius is 15. Given a predicted disparity, the size of the search window is [-2, 2] (in pixels) around the predicted position. Note, however, that the algorithm is not very sensitive to the parameters described here, provided they are within a reasonable range.

Matching and reconstructing the 10,000 points of the synthetic scene (512x512) of section 4 takes about 1mn 20s on a SPARCstation 2 without source code optimization.

Score computation is the most time-consuming part of this algorithm and it is entirely parallelizable. Experiments show that 3 to 5 iterations are sufficient, with about 80% matches validated at the first iteration.

In short, the rough similarity criterion seems reasonable in the presence of noise. Support computation is to some extent similar to the voting mechanism for contour chain matching [**?**, **?**]. Yet our scheme has no restriction at all with respect to primitives. By taking into account the noisy localization in contour detection, a small DG limit (0.2) can be used (compared to 0.5 in [**?**]) and thus a very good disambiguating power is obtained. All these considerations make our algorithm robust but not restrictive. The computation involved is extremely simple and the most time-consuming part is parallelizable.

2.3 Trinocular stereovision using line segments

The two previous algorithms perform binocular stereo. By using three cameras, however, it is possible remove some of the ambiguities and lessen the required amounts of computation.

In this section we present an older but faster trinocular stereovision algorithm [**?**]. This edge-based algorithm establishes correspondences between line segments in the images.

2.3.1 The system of cameras

The system of three cameras has been previously calibrated [**?**] in such a way that the intrinsic and extrinsic parameters of each camera are precisely known. Then, the epipolar geometry can be derived with good accuracy. We also assume that the scene depth lies between two values z_{min} and z_{max}, which define portions of the epipolar lines that we will call *epipolar segments*.

2.3.2 2D primitives

First, edge pixels are obtained by means of a recursive implementation [**?**] of Canny's edge detector. A linking algorithm [**?**] creates chains of connected edge pixels. A polygonal approximation of the chains of pixels is finally performed [**?**], yielding the primitives to be matched.

2.3.3 The matching process

The algorithm works in parallel for each segment of the first image. Its flow of control is described in Figure 2.3.3.

2.3.4 Validation phase

A final validation part discards the triplets that yield too short or isolated 3D segments. In Figure 2.3.4 we show a triplet of images and the reconstructed segments on Figure 2.3.4. Extracting the contours and performing the polygonal approximation for a triplet of 512x512 images takes 40 seconds on a SPARCstation 2 and the matching 10 seconds.

2.4 Trinocular stereovision using curves

This algorithm generalizes the previous one by replacing the polygonal approximation by a cubic B-spline one. A more complete description of the algorithm is given in [**?**].

For each segment S_1 in the first image, whose midpoint is m_1

Compute the epipolar segments L_{12}, L_{13} of m_1 in images 2, 3

For each segment S_2 of image 2 which intersects L_{12} at m_2

Compare attributes of S_1 and S_2 such as orientation, length, intensity gradient

If these attributes are similar, then add S_2 to the list of potential matches of S_1

For each segment S_2 potential match of S_1

Compute the intersection m_3 of L_{13} and L_{23} (issued from m_2 in image 3)

For each segment S_3 in the neighborhood of m_3

Compare the geometric attributes of S_3 with those of S_1, S_2, i.e. compare the orientation of S_3 with the direction predicted from S_1, S_2

If these constraints are verified, then (S_1, S_2, S_3) is a potential match

The segment-based algorithm

5.0cmima/Tea1.ps 5.0cmima/Tea2.ps 5.0cmima/Tea3.ps

Three images of an indoor scene

2.4.1 2D primitives

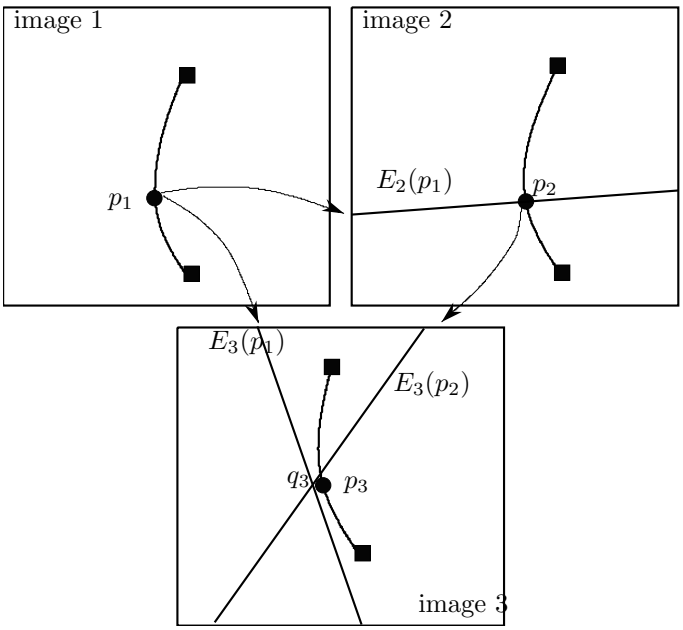
The same assumptions are made about the system of cameras, and the chains of connected edge pixels are obtained in the same manner. Each chain is approximated with two cubic B-splines (one along each coordinate axis). The coefficients, the number of knots and their positions are computed automatically for each spline, by minimizing an energy related to the discontinuity jumps in the third order derivative of the spline function at the knots.

The information provided by a polygonal approximation of the chains of pixels is also used in some parts of the algorithm.

7.5cmaima/LuluFace.ps 7.5cmbima/LuluDessus.ps

14cm(a) The reconstructed 3D segments for the tea box scene of image 2.3.4 projected on the first camera in (a), and seen from above in (b).

2.4.2 Creation of a point triplet



Building a triplet of homologous points

A prediction–verification scheme very similar to the one described in the previous subsubsection is applied. In fig2.4.2 we show how a triplet (p_1, p_2, p_3) of homologous points is built, given the three supporting curves $(\gamma_1, \gamma_2, \gamma_3)$ and the position of p_1 on (γ_1):

- First, we compute the intersection of the epipolar line $E_2(p_1)$ issued from p_1 in the second image with γ_2, which yields p_2.
- Then we compute $q_3 = E_3(p_1) \cap E_3(p_2)$, which constitutes a prediction for the position of p_3.
- Finally, we compute p_3 as the point on γ_3 that minimizes distance to q_3.

The validation of a point triplet also requires some other constraints to be verified:

- Geometric constraints which hold that the prediction–verification scheme can also be applied to the direction of the tangent and the value of the curvature of the curve.

- Binocular heuristic constraints, based on the similarity of the two images, and dealing with geometric attributes such as the direction of the tangent to the curve or the orientation of the intensity gradient.

2.4.3 Creation of a trinocular hypothesis

To build a matching primitive, the principle of the algorithm is, first, to find a triplet of corresponding points on three splines. This is done during the bootstrapping phase, computationally the most expensive:

- A sampling is performed on all the curves of the first image. In practice, we keep one point per segment of the polygonal approximation.

- For each point p_1 of this sampling, all the intersection points p_{2_i} between the epipolar line $E_2(p_1)$ and the curves in image 2 are computed, and for each of them a point p_{3_i} is searched in the third image.

To reduce the complexity of the search, we use the sweeping algorithm described in details in [**?**].

Another way of obtaining point triplets is to run the line-segment algorithm first, and then for each segment triplet, to find a triplet of homologous points lying on the supporting portions of curves, simply using the method described above.

Then comes the propagation phase: for each resulting triplet (p_1, p_2, p_3), we try to propagate along the three supporting curves to find other matching triplets. A point q_1 is chosen on (γ_1), close to p_1 (a finer sampling is used). The method described before is applied on $(\gamma_1, \gamma_2, \gamma_3)$ to find a new triplet including q_1. This is done as long as new point triplets can be found on the same three curves. It finally yields a set of point triplets ordered along $(\gamma_1, \gamma_2, \gamma_3)$, which is called a *trinocular hypothesis*.

2.4.4 Elimination of conflicts between hypotheses

Some of the trinocular hypotheses may be wrong. To eliminate them, we use some of the constraints introduced in section 2.2. We forbid overlapping hypotheses (uniqueness constraint). If two hypotheses overlap, the one with shorter support is discarded. In this

way, we take figural continuity into account. For each remaining hypothesis, a simple least-squares 3D-reconstruction is performed on the point triplets, and issues a set of linked 3D points as shown in Figure 2.4.4.

7.5cmaima/LucFace.ps 7.5cmbima/LucDessus.ps

14cm(a) The reconstructed 3D curves for the tea box scene of image 2.3.4 projected on the first camera in (a), and seen from above in (b).

Extracting the contours and performing the polygonal approximation takes the same 40 seconds on a SPARCstation 2 as in the case of the segment-based algorithm of section 2.3 but matching is slower. In the case of the results shown in Figure 2.4.4, there were about 380 chains of contour in each image and it took 2mn30s to fit the splines and 2mn15s to match them.

3 Evaluating the methods using real data

3.1 Evaluating the correlation-based algorithm

To evaluate the performance of the correlation-based algorithm of section 2.1, we have used scenes such as the one of section 2.1 in which over 200 randomly located points were matched by hand.[3] In these experiments, we have compared the binocular correlation results obtained for various window sizes and the four correlation measures listed below:

$$C_1 = \frac{\sum (I_1 - I_2)^2}{\sqrt{(\sum I_1^2 \sum I_2^2)}} \qquad \text{Non normalized mean-squared differences.}$$

$$C_2 = \frac{\sum I_1 I_2}{\sqrt{(\sum I_1^2 \sum I_2^2)}} \qquad \text{Non normalized cross correlation.}$$

$$C_3 = \frac{\sum ((I_1 - \overline{I_1}) - (I_2 - \overline{I_2}))^2}{\sqrt{(\sum (I_1 - \overline{I_1})^2 \sum (I_2 - \overline{I_2}^2)}} \qquad \text{Normalized mean-squared differences.}$$

$$C_4 = \frac{\sum (I_1 - \overline{I_1})(I_2 - \overline{I_2})}{\sqrt{(\sum (I_1 - \overline{I_1})^2 \sum (I_2 - \overline{I_2})^2)}} \qquad \text{Normalized cross correlation.}$$

In Figure 3.1 we plot the following quantities for each correlation score:

- Percentage of the reference points for which a match has been found.

[3] Fiducial marks have first been pasted on the rocks and then two images for each camera position have been shot, one with the marks and one without them.

- Percentage of the reference points for which the computed disparity is within one pixel the one found by hand.

- Average difference between the computed disparities and the hand-picked ones.

In Figure 3.1 and for each of the correlation measures, we plot the results as functions of the window size. The plots all essentially have the same shape: the percentage of matched points increases with the window size while the precision decreases. By using large windows, we smooth out the finer details and, in effect, reduce the resolution.

Scores C_2, C_3, and C_4 yield results that are very similar and are much better that the ones computed using score C_1. It is easy to understand why in the case of C_3 and C_4; both criteria being normalized, they are insensitive to variations in the mean intensity value of the images that can overwhelm criterion C_1. It is more interesting to note that C_2, even though it is not normalized, is also much better than C_1. By changing the camera settings for one image of the stereo pair, we have checked that the normalized criteria are effectively insensitive to such transformations while C_2 degrades somewhat and C_1 degrades dramatically.

The optimal precision of .66 pixels is obtained for 5x5 windows and is better than one pixel for 7x7 windows with a slightly higher density of points that are matched. In both cases for over 90% of the reference points, the computed disparity is within 1 pixel of the correct one. For a more exhaustive description of these tests, we refer the interested reader to a companion report [**?**].

3.2 Evaluating the contour based algorithms

Evaluating the precision of the contour-based methods on outdoor scenes is more difficult since they do not necessarily give depth information at the reference points we have used above.

In Figure 3.2, we superpose the projected 3D curves extracted from the triplet of Figure 2.1 by the algorithm of section 2.4 on the disparity map produced by the algorithm of section 2.1. We have shifted those contours by the amount predicted by the disparity map and compared the result with their projection in the second image. We have measured average distances of less than a pixel, which indicates that the differences between the

disparities computed using both methods are within that range. The same experiment has been performed using the other contour based methods, yielding the same estimate.

(a)

(b)

(c)

14cmResults as functions of the window size for measures C_1 to C_4: (a) Percentage of reference points for which a match has been found. (b) Percentage of reference points for which the correct match has been found. (c) Average difference between the computed and "real" disparities. The four plots are superposed on the same graphs using the following convention: $C_1 =$, $C_2 =$, $C_3 =$, $C_4 =$.

5cmima/RocksCorrAndSplines.ps

14cm Reprojection of the 3D splines on the disparity map. Note that the correlation-based algorithm tends to widen the rocks by half the window size.

To derive a better quantitative estimate of the relative precisions of the algorithms discussed above, we now turn to synthetic data containing objects of known geometric properties.

4 Evaluating the methods using synthetic data

There are several criteria that can be used to compare stereo algorithms. Two of the most important ones are

1. number of false matches or robustness,

2. accuracy or precision.

To compare our methods and their performance with respects to these criteria, we have used ray-tracing to generate the 3 views of Figure 4. The scene contains a cylinder, a cone, an ashtray, a torus and a calibration grid whose exact positions in space are known. Textures are associated to the first three objects.

Robustness We have applied each of the four stereo algorithms to these images, and obtained four 3-D maps. Note that the correlation algorithm of section 2.1 and the contour-based algorithm of section 2.2 use only two images. Figures 4– 4 show the reconstruction results. In each figure, the left picture is the perspective projection of the 3-D reconstruction by the corresponding algorithm on the first camera, while the right one is the orthographic projection on a horizontal plane which we call the *top view*. This view can be used to judge qualitatively the algorithms; points that have been assigned erro-

5.0cmima/Synt1.ps 5.0cmima/Synt2.ps 5.0cmima/Synt3.ps Three views of a synthetic scene

Subsampled reconstruction results for the correlation algorithm.

neous depths show up clearly as being inconsistent and unrelated with their neighbors. The top views are therefore very useful to evaluate the number of false matches.

Accuracy To gauge the precision of the algorithms, we have chosen 50 arbitrary but well-distributed points on the cylinder, the sphere, and the calibration pattern. For the cylinder, we have compared the distance from the chosen points to the axis with the true radius of the object (200mm) and computed the average error for each of the algorithms. Similarly for the sphere, we have compared the distance from the points to the center with its expected value (200 mm). Finally, for the calibration pattern, we compute the average distances of the reconstructed points to the actual planes. The results are summarized in the table of Figure 4. Overall the reconstruction errors are all less than 1% of the distance. Roughly speaking, each algorithm reconstructs objects that area close to the cameras with a better precision than distant ones. The cone is located at 4318 mm from the first camera. The ashtray is located at almost the same distance, but is lower. The cylinder and the sphere are found at 2943 mm and 2617 mm from the first camera

Reconstruction results for the contour-based binocular algorithm

while the calibration pattern is located at 1480 mm. All the algorithms give a better reconstruction of the calibration pattern than those of the cylinder and the sphere, which in turn are better than those of the cone and ashtray.

Comparison The correlation-based algorithm gives excellent results on the very textured ashtray, sphere and cylinder of Figure 4 but produces numerous false matches on the repetitive patterns of the calibration grid. It does not use the trinocular or hierarchical constraints that the other algorithms bring to bear and that prove essential in this particular case.

The segment-based algorithm yields rather good results for the cylinder and the sphere of Figure 4 because the texture of these objects can be approximated reasonably well by line segments. This approximation gives subpixel precision at edge locations. However, since there is no texture on the cone, the segment-based algorithm performs poorly there.

The curve-based algorithm yields far fewer false matches and allows a much better recovery of the shape of the cone than the segment-based one. The binocular contour-

Reconstruction results for the trinocular algorithm using segments

based one is somewhat less precise but yields even fewer false matches.

5 Conclusion

All the algorithms discussed in this paper yield disparities that are precise to better than a pixel. The quality of the results, however, depends critically on the nature of the edges and textures. For textured scenes such as those of Figures 2.1 and 2.1, the correlation-based algorithm yields denser depth maps but may be computationally more intensive than the contour-based ones. These tend to be more effective for relatively untextured scenes such as the one of Figure 2.3.4. Among them, the algorithm of section 2.2, that uses chains of contours as its primitives performs better than the one of section 2.4 that uses splines for rough contours and worse for smoother objects.

In short, there is no such thing as an "optimal" stereo algorithm. There are only algorithms that are well adapted to the environment they will have to operate in and the primitives that are relevant. Note also that the results shown in this paper depend on

Reconstruction results for the trinocular algorithm using curves

Object / Algorithm	Correlation	Bin. contours	Trin. segments	Trin. curves
Cylinder	4.2 mm	9.7 mm	7.2mm	11.4 mm
Sphere	9.3 mm	16.6 mm	10.9mm	11.2 mm
Grid	3.3 mm	5.8 mm	2.8mm	2.6 mm

Precision of the four algorithms

arbitrary parameters such as a size of the correlation window (section 2.1) or the DG limit (section 2.2). A more thorough study is therefore in order, but designing an appropriate testbed is itself a research topic that we plan to tackle in the future.

We hope, however, that this paper will provide a useful guide to those wishing to use these algorithms and make them cooperate to achieve increased performance.